

## **SGBD - Anul 2**

### **Laborator 0**

---

#### **Recapitulare SQL**

Exercițiile următoare se referă la modelul HR, utilizat în cadrul laboratoarelor din anul 2. Pentru a vă reaminti entitățile, atributele și relațiile, deschideți diagrama corespunzătoare acestui model.

Pentru următoarele exerciții, este necesară și execuția scriptului *hr\_project.sql*. Acesta creează și introduce date în tabelele descrise în continuare.

1) *PROJECT(project\_id#, project\_name, budget, start\_date, deadline, delivery\_date, project\_manager)*

- *project\_id* reprezintă codul proiectului și este cheia primară a relației PROJECT
- *project\_name* reprezintă numele proiectului
- *budget* este bugetul alocat proiectului
- *start\_date* este data demarării proiectului
- *deadline* reprezintă data la care proiectul trebuie să fie finalizat
- *delivery\_date* este data la care proiectul este livrat efectiv
- *project\_manager* reprezintă codul managerului de proiect și este cheie externă. Pe cine referă această coloană și ce relație implementează ea?

2) *WORKS\_ON(project\_id#, employee\_id#, start\_date, end\_date)*

- cheia primară a relației este compusă din atributele *employee\_id* și *project\_id*.

Acest tabel modelează asocierea: "angajat lucreaza în cadrul unui proiect".

#### **Exerciții**

1. Pentru fiecare oraș, să se afișeze țara în care se află și numărul de angajați din cadrul său.
2. Care sunt primii 5 cel mai bine plătiți angajați?
3. Să se obțină numărul de angajați care au mai avut cel puțin trei job-uri, luându-se în considerare și job-ul curent.
4. Cum se numește și în ce țară se află situat departamentul având cei mai mulți angajați?
5. Care este numărul de angajați care sunt şefi?
6. Să se calculeze comisionul mediu din firmă, luând în considerare toate liniile din tabel.
7. Scrieți o cerere pentru a afișa job-ul, salariul total pentru job-ul respectiv pe departamente și salariul total pentru job-ul respectiv pe departamentele 30, 60, 90. Se vor eticheta coloanele corespunzător. Rezultatul va apărea sub forma de mai jos:

Job	Dep30	Dep60	Dep90	Total
-----	-------	-------	-------	-------

---

.....

.....

8. Să se afișeze numele departamentelor, titlurile job-urilor și valoarea medie a salariilor, pentru:
  - fiecare departament și, în cadrul său pentru fiecare job;
  - fiecare departament (indiferent de job);
  - întreg tabelul.
9. Să se listeze angajații (codul și numele acestora) care au lucrat pe cel puțin două proiecte nelivrante la termen.
10. Să se listeze codurile angajaților și codurile proiectelor pe care au lucrat. Listarea va cuprinde și angajații care nu au lucrat pe nici un proiect.
11. Să se afișeze angajații care lucrează în același departament cu cel puțin un manager de proiect.
12. Să se afișeze angajații care nu lucrează în același departament cu nici un manager de proiect.
13. Să se determine departamentele având media salariilor mai mare decât 3000.
14. Informații despre managerii de proiect care au condus 2 proiecte.
15. Lista angajaților care au lucrat numai pe proiecte conduse de managerul de proiect având codul 102.
16. a) Să se obțină numele angajaților care au lucrat **cel puțin** pe aceleași proiecte ca și angajatul având codul 200.  
b) Să se obțină numele angajaților care au lucrat **cel mult** pe aceleași proiecte ca și angajatul având codul 200.

17. Să se obțină angajații care au lucrat pe aceleași proiecte ca și angajatul având codul 200.
18. Să se creeze copiile *EMP\_pnu*, *DEP\_pnu* ale tabelelor *EMPLOYEES*, respectiv *DEPARTMENTS*. Adăugați constrângările de integritate asupra acestor tabele.
19. Introduceți constrângerea ca salariul să nu poată fi *null*. Verificați constrângerea și apoi eliminați-o.

20. Să se introducă în tabelul *EMP\_pnu* (propriu fiecărui utilizator) o linie.

```
INSERT INTO EMP_pnu (employee_id, last_name, email, hire_date,
job_id, salary)
VALUES(...);
```

21. Introduceți un punct intermediar în procesarea tranzacției.

```
SAVEPOINT a;
```

22. Să se șteargă salariatul din tabelul *EMP\_pnu* având salariul minim, introducându-se valoarea acestuia într-o variabilă de legătură. Afipați valoarea acestei variabile.

```
VARIABLE x NUMBER

DELETE FROM employees
WHERE salary = (SELECT MIN(salary)
                 FROM employees)
RETURNING salary INTO :x;

PRINT x;
```

23. Definiți alt punct intermediar în procesarea tranzacției.

SAVEPOINT b;

24. Ștergeți salariul angajatului cel mai apropiat de media salariilor. Procedând analog cu exercițiul 14, afișați codul angajatului actualizat.
25. Reveniți la punctul intermediar de la exercițiul 13. În ce stare se află tabelele?  
S-a încheiat tranzacția ?  
Ce efect are comanda ROLLBACK (la ce stare ne duce)?
26. Să se realizeze o creștere cu 10% a salariului șefului departamentului având cel mai mare număr de angajați.
27. Să se actualizeze, la valoarea implicită (*default*), salariul angajatului având codul maxim în tabelul *emp\_pnu*.
28. a) Să se creeze vizualizarea *VIZ\_EMP\_DEP\_PNU*, pe baza tabelelor *EMP\_PNU* și *DEP\_PNU*, astfel încât aceasta să includă coloanele reprezentând codul angajatului, numele acestuia, email-ul, funcția, data angajării, precum și numele și codul departamentului. Să se introducă alias-uri pentru coloanele vizualizării.  
b) Inserați o linie prin intermediul acestei vizualizări.  
c) Care sunt coloanele actualizabile ale acestei vizualizări? Ce fel de tabel este cel ale cărui coloane sunt actualizabile? Inserați o linie, completând doar valorile corespunzătoare.
29. Să se obțină codurile salariaților atașați tuturor proiectelor pentru care s-a alocat un buget egal cu 10000. (implementarea operatorului DIVISION - 2 metode)
30. Să se listeze informații despre angajații care au lucrat în toate proiectele demarate în primele 6 luni ale anului 2006.
31. Să se listeze informații despre proiectele la care au participat toți angajații care au deținut alte 2 posturi în firmă.